EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2	"20030126367".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:12
L2	47	(Juan near2 Revilla).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:12
L3	43	(Ravi near2 Kolagotla).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:12
L4	86	2 or 3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:12
L5	4	2 and 3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:12
L7	27659	"711"/.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:13
L8	1246	TLD OR DLAT	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:15
L9	5145	TLB OR DLAT	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:15
L10	2354	(translation adj lookaside adj buffer)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:15

EAST Search History

L11	5496	9 or 10	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:15
L12	93995	cache	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:15
L13	46752	USB	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:15
L14	93995	cache	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L15	46752	USB	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L16	862	L14 same L15	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L17	55438	SRAM	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L18	862	L14 same L15	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L19	55438	SRAM	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L20	130	L18 AND L19	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16

EAST Search History

L21	138104	hit or miss	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L22	1527824	index or descriptor$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L23	138104	hit or miss	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L24	1527824	index or descriptor$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L25	22938	L23 and L24	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L26	22938	L23 and L24	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L27	130	L18 AND L19	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L28	22938	L23 and L24	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L29	22	L28 and L27	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L30	104722	hierarchy or hierarchical	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16

EAST Search History

L31	22	L28 and L27	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L32	104722	hierarchy or hierarchical	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16
L33	1	L31 and L32	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/01/19 14:16

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ◉ The ACM Digital Library ○ The Guide

+hierarchical +cache, +TLB, +cache +miss, +local +address h SEARCH

Feedback Report a problem Satisfaction survey

Terms used
hierarchical cache TLB cache miss local address hit or miss cross interrogate

Found **122** of **169,866**

Sort results by	relevance	Save results to a Binder
Display results	expanded form	Search Tips
		☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 122   Result page: **1** 2 3 4 5 6 7 next

Relevance scale

1 RISCY patents
David A. Patterson
September 1988 **ACM SIGARCH Computer Architecture News**, Volume 16 Issue 4
Publisher: ACM Press
Full text available: pdf(1.83 MB)   Additional Information: full citation, index terms

2 UTLB: a mechanism for address translation on network interfaces
Yuqun Chen, Angelos Bilas, Stefanos N. Damianakis, Cezary Dubnicki, Kai Li
October 1998 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the eighth international conference on Architectural support for programming languages and operating systems ASPLOS-VIII**, Volume 33 , 32 Issue 11 , 5
Publisher: ACM Press
Full text available: pdf(1.76 MB)   Additional Information: full citation, abstract, references, citings, index terms

An important aspect of a high-speed network system is the ability to transfer data directly between the network interface and application buffers. Such a *direct data path* requires the network interface to "know" the virtual-to-physical address translation of a user buffer, *i.e.*, the physical memory location of the buffer. This paper presents an efficient address translation architecture, User-managed TLB (UTLB), which eliminates system calls and device interrupts from the common co ...

3 Organization and performance of a two-level virtual-real cache hierarchy
W. H. Wang, J.-L. Baer, H. M. Levy
April 1989 **ACM SIGARCH Computer Architecture News , Proceedings of the 16th annual international symposium on Computer architecture ISCA '89**, Volume 17 Issue 3
Publisher: ACM Press
Full text available: pdf(1.01 MB)   Additional Information: full citation, abstract, references, citings, index terms

We propose and analyze a two-level cache organization that provides high memory bandwidth. The first-level cache is accessed directly by virtual addresses. It is small, fast, and, without the burden of address translation, can easily be optimized to match the processor speed. The virtually-addressed cache is backed up by a large physically-addressed cache; this second-level cache provides a high hit ratio and greatly reduces memory traffic. We show how the second-level cache can be easily e ...

4 Architectural support for translation table management in large address space machines
Jerry Huck, Jim Hays
May 1993 **ACM SIGARCH Computer Architecture News , Proceedings of the 20th annual international symposium on Computer architecture ISCA '93**, Volume 21 Issue 2
Publisher: ACM Press
Full text available: pdf(1.34 MB) Additional Information: full citation, abstract, references, citings, index terms

Virtual memory page translation tables provide mappings from virtual to physical addresses. When the hardware controlled Translation Lookaside Buffers (TLBs) do not contain a translation, these tables provide the translation. Approaches to the structure and management of these tables vary from full hardware implementations to complete software based algorithms. The size of the virtual address space used by processes is rapidly growing beyond 32 bits of address. As the utilized ad ...

5 A large, fast instruction window for tolerating cache misses
Alvin R. Lebeck, Jinson Koppanalil, Tong Li, Jaidev Patwardhan, Eric Rotenberg
May 2002 **ACM SIGARCH Computer Architecture News , Proceedings of the 29th annual international symposium on Computer architecture ISCA '02 , Proceedings of the 29th annual international symposium on Computer architecture ISCA '02**, Volume 30 Issue 2
Publisher: IEEE Computer Society, ACM Press
Full text available: pdf(1.22 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Instruction window size is an important design parameter for many modern processors. Large instruction windows offer the potential advantage of exposing large amounts of instruction level parallelism. Unfortunately naively scaling conventional window designs can significantly degrade clock cycle time, undermining the benefits of increased parallelism.This paper presents a new instruction window design targeted at achieving the latency tolerance of large windows with the clock cycle time of small ...

Keywords: Instruction Window, Memory Latency, Cache Memory, Latency Tolerance

6 SoftFLASH: analyzing the performance of clustered distributed virtual shared memory
Andrew Erlichson, Neal Nuckolls, Greg Chesson, John Hennessy
September 1996 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the seventh international conference on Architectural support for programming languages and operating systems ASPLOS-VII**, Volume 31 , 30 Issue 9 , 5
Publisher: ACM Press
Full text available: pdf(1.29 MB) Additional Information: full citation, abstract, references, citings, index terms

One potentially attractive way to build large-scale shared-memory machines is to use small-scale to medium-scale shared-memory machines as clusters that are interconnected with an off-the-shelf network. To create a shared-memory programming environment across the clusters, it is possible to use a virtual shared-memory software layer. Because of the low latency and high bandwidth of the interconnect available within each cluster, there are clear advantages in making the clusters as large as possi ...

7 MGS: a multigrain shared memory system
Donald Yeung, John Kubiatowicz, Anant Agarwal
May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture ISCA '96**, Volume 24 Issue 2
Publisher: ACM Press
Full text available: pdf(1.37 MB) Additional Information: full citation, abstract, references, citings, index terms

Parallel workstations, each comprising 10-100 processors, promise cost-effective general-purpose multiprocessing. This paper explores the coupling of such small- to medium-scale shared memory multiprocessors through software over a local area network to synthesize larger shared memory systems. We call these systems Distributed Scalable Shared-memory Multiprocessors (DSSMPs).This paper introduces the design of a shared memory system that uses multiple granularities of sharing, and presents an imp ...

8 System-level power optimization: techniques and tools
Luca Benini, Giovanni de Micheli
April 2000 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 5 Issue 2
Publisher: ACM Press
Full text available: pdf(385.22 KB) Additional Information: full citation, abstract, references, citings, index terms

This tutorial surveys design methods for energy-efficient system-level design. We consider electronic sytems consisting of a hardware platform and software layers. We consider the three major constituents of hardware that consume energy, namely computation, communication, and storage units, and we review methods of reducing their energy consumption. We also study models for analyzing the energy cost of software, and methods for energy-efficient software design and compilation. This survery ...

9 The DASH prototype: implementation and performance
Daniel Lenoski, James Laudon, Truman Joe, David Nakahira, Luis Stevens, Anoop Gupta, John Hennessy
August 1998 **25 years of the international symposia on Computer architecture (selected papers)**
Publisher: ACM Press
Full text available: pdf(1.52 MB) Additional Information: full citation, references, index terms

10 Automatic tiling of iterative stencil loops
Zhiyuan Li, Yonghong Song
November 2004 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 26 Issue 6
Publisher: ACM Press
Full text available: pdf(947.69 KB) Additional Information: full citation, abstract, references, index terms

Iterative stencil loops are used in scientific programs to implement relaxation methods for numerical simulation and signal processing. Such loops iteratively modify the same array elements over different time steps, which presents opportunities for the compiler to improve the temporal data locality through loop tiling. This article presents a compiler framework for automatic tiling of iterative stencil loops, with the objective of improving the cache performance. The article first presents a ...

Keywords: Caches, loop transformations, optimizing compilers

11 The DASH prototype: implementation and performance
Daniel Lenoski, James Laudon, Truman Joe, David Nakahira, Luis Stevens, Anoop Gupta, John Hennessy
April 1992 **ACM SIGARCH Computer Architecture News , Proceedings of the 19th annual international symposium on Computer architecture ISCA '92**, Volume 20 Issue 2
Publisher: ACM Press
Full text available: pdf(1.68 MB) Additional Information: full citation, abstract, references, citings, index terms

The fundamental premise behind the DASH project is that it is feasible to build large-scale shared-memory multiprocessors with hardware cache coherence. While paper studies and software simulators are useful for understanding many high-level design trade-offs, prototypes are essential to ensure that no critical details are overlooked. A prototype provides convincing evidence of the feasibility of the design allows one to accurately estimate both the hardware and the complexity cost of vario ...

12 Measuring Experimental Error in Microprocessor Simulation
Rajagopalan Desikan, Doug Burger, Stephen W. Keckler
June 2001 **Proceedings of the 28th annual international symposium on Computer architecture**
Publisher: ACM Press
Full text available: pdf(237.69 KB) Additional Information: full citation, abstract, citings, index terms

Abstract: We measure the experimental error that arises from the use of non-validated simulators in computer architecture research, with the goal of increasing the rigor of simulation- based studies. We describe the methodology that we used to validate a microprocessor simulator against a Compaq DS-10L workstation, which contains an Alpha 21264 processor. Our evaluation suite consists of a set of 21 microbenchmarks that stress different aspects of the 21264 microarchitecture. Using the microbenc ...

13 Measuring experimental error in microprocessor simulation
Rajagopalan Desikan, Doug Burger, Stephen W. Keckler
May 2001 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 2001 symposium on Software reusability: putting software reuse in context SSR '01**, Volume 26 Issue 3
Publisher: ACM Press
Full text available: pdf(1.03 MB) Additional Information: full citation, references, index terms

14 Interconnections in Multi-Core Architectures: Understanding Mechanisms, Overheads and Scaling
Rakesh Kumar, Victor Zyuban, Dean M. Tullsen
May 2005 **ACM SIGARCH Computer Architecture News , Proceedings of the 32nd Annual International Symposium on Computer Architecture ISCA '05**, Volume 33 Issue 2
Publisher: IEEE Computer Society, ACM Press
Full text available: pdf(235.90 KB) Additional Information: full citation, abstract, index terms

This paper examines the area, power, performance, and design issues for the on-chip interconnects on a chip multiprocessor, attempting to present a comprehensive view of a class of interconnect architectures. It shows that the design choices for the interconnect have significant effect on the rest of the chip, potentially consuming a significant fraction of the real estate and power budget. This research shows that designs that treat interconnect as an entity that can be independently architecte ...

15 The performance of μ-kernel-based systems
Hermann Härtig, Michael Hohmuth, Jochen Liedtke, Sebastian Schönberg
October 1997 **ACM SIGOPS Operating Systems Review , Proceedings of the sixteenth ACM symposium on Operating systems principles SOSP '97**, Volume 31 Issue 5
Publisher: ACM Press
Full text available: pdf(2.02 MB) Additional Information: full citation, references, citings, index terms

16 Comparative performance evaluation of cache-coherent NUMA and COMA architectures
Per Stenström, Truman Joe, Anoop Gupta
April 1992 **ACM SIGARCH Computer Architecture News , Proceedings of the 19th annual international symposium on Computer architecture ISCA '92**, Volume 20 Issue 2
Publisher: ACM Press
Full text available: pdf(1.52 MB) Additional Information: full citation, abstract, references, citings, index terms

Two interesting variations of large-scale shared-memory machines that have recently emerged are cache-coherent non-uniform-memory-access machines (CC-NUMA) and cache-only memory architectures (COMA). They both have distributed main memory and use directory-based cache coherence. Unlike CC-NUMA, however, COMA machines automatically migrate and replicate data at the main-memory level in cache-line sized chunks. This paper compares the performance of these two classes ...

17 Tuning compiler optimizations for simultaneous multithreading
Jack L. Lo, Susan J. Eggers, Henry M. Levy, Sujay S. Parekh, Dean M. Tullsen
December 1997 **Proceedings of the 30th annual ACM/IEEE international symposium on Microarchitecture**
Publisher: IEEE Computer Society
Full text available: pdf(1.45 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Compiler optimizations are often driven by specific assumptions about the underlying architecture and implementation of the target machine. For example, when targeting shared-memory multiprocessors, parallel programs are compiled to minimize sharing, in order to decrease high-cost, inter-processor communication. This paper reexamines several compiler optimizations in the context of simultaneous multithreading (SMT), a processor architecture that issues instructions from multiple threads to the f ...

Keywords: cache size, compiler optimizations, cyclic algorithm, fine-grained sharing, instructions, inter-processor communication, inter-thread instruction-level parallelism, latency hiding, loop tiling, loop-iteration scheduling, memory system resources, optimising compilers, parallel architecture, parallel programs, performance, processor architecture, shared-memory multiprocessors, simultaneous multithreading, software speculative execution

18 Multi-level texture caching for 3D graphics hardware
Michael Cox, Narendra Bhandari, Michael Shantz
April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture ISCA '98**, Volume 26 Issue 3
Publisher: IEEE Computer Society, ACM Press
Full text available: Additional Information: full citation, abstract, references, citings, index

pdf(1.62 MB) Publisher Site   terms

Traditional graphics hardware architectures implement what we call the *push architecture* for texture mapping. Local memory is dedicated to the accelerator for fast local retrieval of texture during rasterization, and the application is responsible for managing this memory. The push architecture has a bandwidth advantage, but disadvantages of limited texture capacity, escalation of accelerator memory requirements (and therefore cost), and poor memory utilization. The push architecture also ...

19 Computing curricula 2001
September 2001 **Journal on Educational Resources in Computing (JERIC)**
Publisher: ACM Press
Full text available: pdf(613.63 KB) html(2.78 KB)   Additional Information: full citation, references, citings, index terms

20 A memory management unit and cache controller for the MARS system
Feipei Lai, Chyuan-Yow Wu, Tai-Ming Parng
November 1990 **Proceedings of the 23rd annual workshop and symposium on Microprogramming and microarchitecture**
Publisher: IEEE Computer Society Press
Full text available: pdf(1.07 MB)   Additional Information: full citation, abstract, references

For large caches, the interaction between cache access and address translation affects the machine cycle time and the access time to memory. The physically addressed caches slow down the cache access due to the virtual address translation. The virtually addressed caches is faster, but the synonym problem is difficult to handle. By some software constraints and hardware support, our virtually addressed physically tagged caches can achieve the same speed as traditional virtually addressed cac ...

Results 1 - 20 of 122   Result page: **1** 2 3 4 5 6 7 next

IEEE Xplore®
RELEASE 2.1

Home | Login | Logout | Access Information | Ale

Welcome United States Patent and Trademark Office

Search Session History

BROWSE SEARCH IEEE XPLORE GUIDE

Thu, 19 Jan 2006, 2:24:30 PM EST

Edit an existing query or compose a new query in the Search Query Display.

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Search Query Display

Recent Search Queries

#1	((hierarchical cache)<in>metadata)
#2	((translation lookaside buffer) or TLB<IN>metadata)
#3	((cache miss) or (cache hit)<IN>metadata)
#4	(local addressable memory<IN>metadata)
#5	(((hierarchical cache)<in>metadata)) <AND> (((translation lookaside buffer) or TLB<IN>metadata))
#6	cache interrogate
#7	(cache protection lookaside buffer) or CPLB
#8	(SRAM or DRAM<IN>metadata)
#9	((SRAM or DRAM<IN>metadata)) <AND> (((translation lookaside buffer) or TLB<IN>metadata))

Help Contact Us Privac

Indexed by
Inspec®